

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:)
)
Masahito KUBO, et al.)
) Group Art Unit: Unassigned
Serial No.: To be assigned)
) Examiner: Unassigned
Filed: March 15, 2001)
)
For: TIMER ADJUSTING SYSTEM)



dlg
C. Halsey
5-11-01

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. §1.55**

*Assistant Commissioner for Patents
Washington, D.C. 20231*

Sir:

In accordance with the provisions of 37 C.F.R. §1.55, the applicant submits herewith a certified copy of the following foreign application:

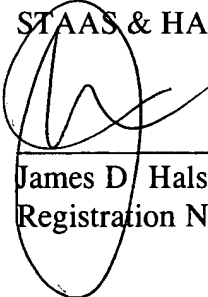
Japanese Patent Application No. 2000-233811
Filed: August 2, 2000.

It is respectfully requested that the applicant be given the benefit of the foreign filing date as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. §119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: March 14, 2001

By: 
James D. Halsey, Jr.
Registration No. 22,729

700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001
(202) 434-1500

PATENT OFFICE
JAPANESE GOVERNMENT

j1046 U.S. PTO
09/808343
03/15/01

This is to certify that the annexed is a true copy of the following
application as filed with this Office.

Date of Application: August 2, 2000

Application Number: Patent Application
No. 2000-233811

Applicant(s): FUJITSU LIMITED

December 1, 2000

Commissioner,
Patent Office Kozo Oikawa

Certificate No. 2000-3098179

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application:

2 0 0 0 年 8 月 2 日

出 願 番 号
Application Number:

特 願 2 0 0 0 - 2 3 3 8 1 1

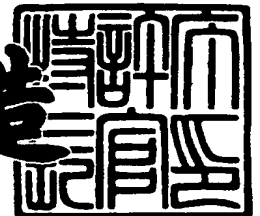
出 願 人
Applicant (s):

富士通株式会社

2 0 0 0 年 1 2 月 1 日

特 許 庁 長 官
Commissioner,
Patent Office

及 川 耕 造



出 証 番 号 出 証 特 2 0 0 0 - 3 0 9 8 1 7 9

【書類名】 特許願

【整理番号】 0000723

【提出日】 平成12年 8月 2日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/12
G06F 15/177

【発明の名称】 タイマ調整システム

【請求項の数】 4

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 久保 雅史

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 上埜 治彦

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 仲川 明和

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100074099

 【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F

 【弁理士】

 【氏名又は名称】 大菅 義之

 【電話番号】 03-3238-0031

【選任した代理人】

【識別番号】 100067987

【住所又は居所】 神奈川県横浜市鶴見区北寺尾 7 - 2 5 - 2 8 - 5 0 3

【弁理士】

【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 タイマ調整システム

【特許請求の範囲】

【請求項 1】 マルチプロセッサシステムにおける複数のプロセッサのタイマを調整するタイマ調整システムであって、

前記マルチプロセッサシステムの時刻同期信号を発生する発生手段と、

前記時刻同期信号を出力する出力手段と、

前記出力手段から出力され、前記マルチプロセッサシステム内を伝播して戻ってきた前記時刻同期信号を、入力する入力手段と、

前記出力手段が前記時刻同期信号を出力してから前記入力手段が該時刻同期信号を入力するまでの時間を測定する測定手段と、

測定された時間を、前記時刻同期信号が前記マルチプロセッサシステム内の 2 つのプロセッサ間を伝播する時間として用いて、前記複数のプロセッサのタイマのうち少なくとも 1 つ以上のタイマの時刻情報を補正し、該複数のプロセッサのタイマを同期させる同期手段と

を備えることを特徴とするタイマ調整システム。

【請求項 2】 前記発生手段は、前記複数のプロセッサのタイマのうちの 1 つが生成する信号を用いて、前記時刻同期信号を発生することを特徴とする請求項 1 記載のタイマ調整システム。

【請求項 3】 前記測定手段は、前記複数のプロセッサのタイマのうちの 1 つを用いて前記時刻同期信号が伝播する時間を測定することを特徴とする請求項 1 記載のタイマ調整システム。

【請求項 4】 マルチプロセッサシステムにおける複数のプロセッサのタイマを調整するタイマ調整システムであって、

前記マルチプロセッサシステムの時刻同期信号を発生する発生手段と、

前記時刻同期信号を同期出力として出力する出力手段と、

同期入力を入力する入力手段と

を前記複数のプロセッサの各々に設け、

前記複数のプロセッサから出力される複数の同期出力の論理和信号を生成し、

該論理和信号を前記同期入力として前記複数のプロセッサに分配する分配手段と

、
前記複数のプロセッサのうちの 1 つが前記同期出力を出力してから前記同期入力を受け取るまでの時間を測定する測定手段と、

測定された時間を、前記論理和信号が伝播する時間として用いて、前記複数のプロセッサのタイマのうち少なくとも 1 つ以上のタイマの時刻情報を補正し、該複数のプロセッサのタイマを同期させる同期手段と

を備えることを特徴とするタイマ調整システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、計算機システムの時刻管理およびタイミングサービスに係り、特にマルチプロセッサシステムにおいてプロセッサ間の時刻調整を行うタイマ調整システムに関する。

【0002】

【従来の技術】

マイクロプロセッサは、一般に、プロセッサの動作クロックにより更新されるタイマ (T I C K) を持っており、このタイマが計算機システムの時刻管理とタイミングサービスに使用されていることが多い。

【0003】

一方、今日の計算機システムは大規模化しており、マルチプロセッサシステムが一般化している。マルチプロセッサシステムでは、システム全体を複数のパーティションに分割して、各パーティション毎に対称型マルチプロセッサシステム (Symmetric Multi-Processors, SMP) として動作させることが多い。対称型マルチプロセッサシステムでは、すべてのプロセッサが共通の主記憶を介して連携しながら処理を行う。

【0004】

このようなシステムでは、同一パーティション内のプロセッサ間で、時刻がある程度一致している必要がある。時刻が一致していなければ、プロセスがプロセ

ッサ間を移動するときに、プロセスの順序関係が逆転する等の問題を引き起こす可能性がある。

【0005】

このため、従来のマルチプロセッサシステムでは、何らかの手段を用いて時刻を表すタイマ値（T I C K 値）を調整し、プロセッサ間でその値を一致させるための工夫がなされている。例えば、ソフトウェアによる時刻調整によれば、プログラムにより、あるプロセッサから T I C K 値が読み出され、主記憶経由で同一パーティション内の他のプロセッサに送られる。そして、T I C K 値を受信したプロセッサは、その値を自分の T I C K に書き込むことで、時刻を変更する。

【0006】

【発明が解決しようとする課題】

しかしながら、上述した従来の時刻調整方法には、次のような問題がある。

半導体技術およびプロセッサ設計技術の進歩により、マイクロプロセッサの動作周波数が 1 G H z を越える時代を迎えつつある。このような昨今の動作周波数の増大に伴い、対称型マルチプロセッサシステムにおいては、T I C K 値の主記憶経由の転送に要する時間が、相対的に無視できない大きさとなってきた。

【0007】

しかし、従来のソフトウェアでは、この転送時間を測定することができないため、その影響を無視して時刻調整を行っている。このため、動作周波数が増大するにつれて、T I C K 値の転送による時刻調整の誤差が、プロセスの順序関係等に影響を与える可能性が高くなる。

【0008】

本発明の課題は、マルチプロセッサシステムにおいて、プロセッサ間の時刻調整をより正確に行うタイマ調整システムを提供することである。

【0009】

【課題を解決するための手段】

図 1 は、本発明のタイマ調整システムの原理図である。本発明の第 1 の局面において、タイマ調整システムは、発生手段 1 1、出力手段 1 2、入力手段 1 3、測定手段 1 4、および同期手段 1 5 を備え、マルチプロセッサシステムにおける

複数のプロセッサのタイマを調整する。

【0010】

発生手段11は、マルチプロセッサシステムの時刻同期信号を発生し、出力手段12は、時刻同期信号を出力し、入力手段13は、出力手段12から出力され、マルチプロセッサシステム内を伝播して戻ってきた時刻同期信号を、入力する。

【0011】

測定手段14は、出力手段12が時刻同期信号を出力してから入力手段13が時刻同期信号を入力するまでの時間を測定する。同期手段15は、測定された時間を、時刻同期信号がマルチプロセッサシステム内の2つのプロセッサ間を伝播する時間として用いて、複数のプロセッサのタイマのうち少なくとも1つ以上のタイマの時刻情報を補正し、それらのプロセッサのタイマを同期させる。

【0012】

発生手段11が発生した時刻同期信号は、出力手段12から出力され、マルチプロセッサシステム内の所定の経路を伝播して、入力手段13に戻ってくる。測定手段14は、時刻同期信号の伝播時間を測定して、同期手段15に渡す。同期手段15は、受け取った伝播時間を2つのプロセッサ間の伝播時間として用いて、補正対象のタイマの時刻情報を補正することで、時刻を調整する。これにより、複数のプロセッサのタイマが同期する。

【0013】

このようなタイマ調整システムによれば、2つのプロセッサ間で時刻同期信号を伝達することができ、その信号の伝播時間を測定することができる。これにより、伝播時間を考慮して各プロセッサのタイマを調整することが可能となり、時刻調整の精度が向上する。

【0014】

また、本発明の第2の局面において、タイマ調整システムは、発生手段11、出力手段12、入力手段13、測定手段14、同期手段15、および分配手段16を備え、マルチプロセッサシステムにおける複数のプロセッサのタイマを調整する。発生手段11、出力手段12、および入力手段13は、複数のプロセッサ

の各々に設けられる。

【 0 0 1 5 】

発生手段 1 1 は、マルチプロセッサシステムの時刻同期信号を発生し、出力手段 1 2 は、時刻同期信号を同期出力として出力し、入力手段 1 3 は、同期入力を入力する。分配手段 1 6 は、複数のプロセッサから出力される複数の同期出力の論理和信号を生成し、その論理和信号を同期入力として複数のプロセッサに分配する。

【 0 0 1 6 】

測定手段 1 4 は、複数のプロセッサのうちの 1 つが同期出力を出力してから同期入力を受け取るまでの時間を測定する。同期手段 1 5 は、測定された時間を、論理和信号が伝播する時間として用いて、複数のプロセッサのタイマのうち少なくとも 1 つ以上のタイマの時刻情報を補正し、それらのプロセッサのタイマを同期させる。

【 0 0 1 7 】

分配手段 1 6 により、複数のプロセッサから出力される同期出力の論理和がそれらのプロセッサに分配されるので、いずれかのプロセッサが同期出力をアサートすれば、すべてのプロセッサの同期入力がアサートされる。したがって、1 つのプロセッサの時刻同期信号を、他のすべてのプロセッサに同時に伝達することができる。

【 0 0 1 8 】

このようなタイマ調整システムによれば、第 1 の局面のタイマ調整システムと同様に、時刻調整の精度が向上するとともに、複数のプロセッサのタイマを容易に調整することができる。

【 0 0 1 9 】

例えば、図 1 の発生手段 1 1 は、後述する図 3 の T I C K レジスタ 5 3、インクリメンタ 5 4、マルチプレクサ 5 5、および A N D 回路 5 6 に対応し、出力手段 1 2 は、A N D 回路 5 6 に対応する。また、入力手段 1 3 は、図 3 の A N D 回路 4 6 に対応し、測定手段 1 4 は、図 3 の T I C K 調整回路と、後述する図 2 の C P U (中央処理装置) 2 1 および主記憶 2 4 に対応する。また、同期手段 1 5

は、図 2 の CPU 2 1 および主記憶 2 4 に対応し、分配手段 1 6 は、図 2 の TICK 調整 Tree 2 6、2 7 に対応する。

【 0 0 2 0 】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態のマルチプロセッサシステムは、各種タイミングサービスを提供するために、複数のタイマをシステム内に分散して持ち、それらのタイマ値を調整して一致させる。タイマ調整のために用いられるシステムハードウェアおよびソフトウェアの概要は、以下の通りである。

(1) システム内の各プロセッサは、システム時刻同期信号として、TICK 同期出力 (TICK. sync-out) と TICK 同期入力 (TICK. sync-in) を備える。

(2) システムハードウェアは、同一パーティション内の全プロセッサが出力する TICK 同期出力を受信し、それらの TICK 同期出力の論理和を、そのパーティション内のプロセッサの時刻同期信号として、全プロセッサに同時に分配する。システムハードウェアが各プロセッサに分配する信号の伝播遅延時間は、システム依存であるが、同一パーティション内ではどのプロセッサに対しても同等とする。

(3) 各プロセッサは、システムハードウェアから時刻同期信号を TICK 同期入力として受信し、自プロセッサの TICK 値の調整に使用する。

(4) 各プロセッサは、ソフトウェアにより時刻同期信号の伝播遅延時間を測定する。ソフトウェアは TICK 同期動作に先立って、伝播遅延時間を測定し、測定された時間を TICK 値の調整に使用する。

【 0 0 2 1 】

このようなシステムによれば、各プロセッサは、ハードウェアによりシステム時刻同期信号を他のプロセッサに送るとともに、その信号の伝播遅延時間を測定することができる。測定された伝播遅延時間を考慮して各プロセッサの TICK 値を調整することで、時刻調整の精度が向上する。

【 0 0 2 2 】

図2は、このようなマルチプロセッサシステムの構成図である。図2のマルチプロセッサシステムは、複数のCPU（中央処理装置、プロセッサ）21、システムバス22、スイッチ回路（例えば、クロスバススイッチ）23、主記憶24、パーティション制御回路25、およびTICK調整Tree 26、27を備える。

【0023】

伝播遅延時間を測定し、TICK同期処理を行う時刻調整プログラムは、例えば、主記憶24に格納され、CPU21により実行される。あるいは、システムバス22上にROM（Read Only Memory）を設けておき、そのROMに時刻調整プログラムを格納してもよい。

【0024】

各CPU21は、システムバス22とスイッチ回路23を介して主記憶24にアクセスする。TICK調整Tree 26、27は、システム共通資源として扱われ、同一パーティション内のTICK同期出力の論理和演算を行い、演算結果をそのパーティションのTICK同期入力として分配する回路を含む。

【0025】

TICK調整Tree 26は、CPU21から信号線31を介して入力されるTICK. sync-outの論理和を求め、信号線32または33を介して、CPU21またはTICK調整Tree 27に出力する。信号線32からCPU21に入力される信号は、TICK. sync-inに対応する。

【0026】

TICK調整Tree 27は、TICK調整Tree 26から信号線33を介して入力される信号の論理和を求め、信号線34を介してTICK調整Tree 26に出力する。信号線34からTICK調整Tree 26に入力される信号は、TICK. sync-inとしてCPU21に出力される。

【0027】

パーティション制御回路25は、システムの論理構成を管理するためのレジスタを含み、システムを複数のパーティションに分割して、各パーティションをそれぞれ別々の対称型マルチプロセッサシステムとして動作させるための制御を行

う。この場合、各パーティションは、それぞれ異なるオペレーティングシステム（OS）により管理される。ここでは、2つのパーティション#0、#1が論理的に生成されており、各パーティションに属する複数のCPU21は、主記憶24の特定の共通領域を使用しながら連携動作を行う。

【0028】

また、パーティション制御回路25は、パーティション毎に別々にタイマを同期させるために、TICK調整Tree26、27をパーティションに基づいて分割する制御も行う。パーティション制御回路25とTICK調整Tree26、27の具体例については、後述することにする。

【0029】

図3は、図2のCPU21内に設けられるTICK調整回路の構成図である。図3のTICK調整回路は、マルチプレクサ40、41、55、TICK同期制御レジスタ42、インバータ回路43、44、47、49、52、AND回路45、46、50、51、56、NAND回路48、TICKレジスタ53、およびインクリメンタ54を含む。

【0030】

このうち、TICKレジスタ53、インクリメンタ54、およびマルチプレクサ55は、タイマ回路を構成する。TICKレジスタ53は、CPU内の時刻を表すTICK値を格納し、インクリメンタ54は、TICKレジスタ53から出力されるTICK値に1を加算して、加算結果を出力する。

【0031】

マルチプレクサ55は、インクリメンタ54の出力と、TICKレジスタ53の出力と、TICKレジスタ53への書き込み値TICK-wr[n:0]のうちの1つを選択して、TICKレジスタ53に出力する。TICK-wr[n:0]の値は、時刻調整プログラムによる制御に基づいてCPU内のALU（Arithmetic and Logic Unit）により生成される。

【0032】

TICK同期制御レジスタ42は、TICK_sync（1ビット）とTICK_adj[1:0]（2ビット）の3つの制御ビットを有する。インバータ回

路43は、TICK同期制御レジスタ42から出力されるTICK. adj [1:0]の値を反転して、マルチプレクサ41に出力する。また、インバータ回路44は、TICK同期制御レジスタ42から出力されるTICK. syncの値を反転して、マルチプレクサ40に出力する。

【0033】

マルチプレクサ40は、TICK同期制御レジスタ42からのTICK. syncと、TICK同期制御レジスタ42への書き込み値TS0-wr [0]のうちの1つを選択して、TICK同期制御レジスタ42のTICK. syncに出力する。

【0034】

マルチプレクサ41は、TICK同期制御レジスタ42からのTICK. adj [1:0]と、TICK同期制御レジスタ42への書き込み値TS1-wr [1:0]と、インバータ回路43の出力のうちの1つを選択して、TICK同期制御レジスタ42のTICK. adj [1:0]に出力する。TS0-wr [0]およびTS1-wr [1:0]の値は、時刻調整プログラムによる制御に基づいてALUにより生成される。

【0035】

AND回路56は、TICK同期制御レジスタ42から出力されるTICK. syncと、インクリメンタ54から出力される信号carry-outの論理積を求め、TICK. sync-outとして出力する。carry-outは、タイマ回路のキャリー出力に対応し、インクリメンタ54においてTICK値の所定ビットの桁上げが発生したときに、論理“1”となる。

【0036】

インバータ回路44は、信号*TS0-wrの値を反転して、マルチプレクサ40に出力する。*TS0-wrは、TICK同期制御レジスタ42に対するTICK. syncの書き込みを指示するための信号TS0-wrの否定に対応する。

【0037】

マルチプレクサ40は、*TS0-wrが論理“1”のとき、TICK. sy

ncを選択し、*TS0-wrが論理“0”（インバータ回路44の出力が論理“1”）のとき、TS0-wr[0]を選択して、選択した値をTICK. sync ncに書き込む。

【0038】

AND回路45、46、インバータ回路47、およびNAND回路48は、マルチプレクサ41の制御信号を生成する。NAND回路48は、TICK. adj[1:0]のビット0の値とTICK. sync-inの論理積の否定を求め、AND回路45に出力する。AND回路45は、NAND回路48の出力と信号*TS1-wrの論理積を求め、マルチプレクサ41に出力する。*TS1-wrは、TICK同期制御レジスタ42に対するTICK. adj[1:0]の書き込みを指示するための信号TS1-wrの否定に対応する。

【0039】

AND回路46は、TICK. adj[1:0]のビット0の値とTICK. sync-inと*TS1-wrの論理積を求め、マルチプレクサ41に出力する。また、インバータ回路47は、*TS1-wrの値を反転して、マルチプレクサ41に出力する。

【0040】

マルチプレクサ41は、AND回路45の出力が論理“1”のとき、TICK. adj[1:0]を選択し、AND回路46の出力が論理“1”のとき、インバータ回路43の出力を選択する。また、インバータ回路47の出力が論理“1”のとき、TS1-wr[1:0]を選択する。

【0041】

*TS1-wrが論理“0”のときは、AND回路45、46の出力が論理“0”となり、インバータ回路47の出力が論理“1”となるため、TS1-wr[1:0]の値がTICK. adj[1:0]に書き込まれる。また、*TS1-wrが論理“1”のときは、インバータ回路47の出力が論理“0”となるため、TICK. adj[1:0]のビット0とTICK. sync-inの値に応じて、TICK. adj[1:0]の設定値が決められる。

【0042】

また、インバータ回路49、52、およびAND回路50、51は、マルチプレクサ55の制御信号を生成する。AND回路50は、 $TICK_adj[1:0]$ のビット1の値と信号 $*TICK_wr$ の論理積を求め、信号 $stop$ としてマルチプレクサ55に出力する。 $*TICK_wr$ は、 $TICK$ レジスタ53への書き込みを指示するための信号 $TICK_wr$ の否定に対応する。

【0043】

インバータ回路49は、 $TICK_adj[1:0]$ のビット1の値を反転して、AND回路51に出力する。AND回路51は、インバータ回路49の出力と信号 $*TICK_wr$ の論理積を求め、信号 $count$ としてマルチプレクサ55に出力する。また、インバータ回路52は、 $*TICK_wr$ の値を反転し、信号 $TICK_write$ としてマルチプレクサ55に出力する。

【0044】

マルチプレクサ55は、 $stop$ が論理“1”のとき、 $TICK$ 値を選択し、 $count$ が論理“1”のとき、インクリメンタ54のカウント値を選択し、 $TICK_write$ が論理“1”のとき、 $TICK_wr[n:0]$ を選択する。

【0045】

$*TICK_wr$ が論理“0”のときは、 $stop$ と $count$ が論理“0”となり、 $TICK_write$ が論理“1”となるため、 $TICK_wr[n:0]$ の値が $TICK$ に書き込まれる。また、 $*TICK_wr$ が論理“1”のときは、 $TICK_write$ が論理“0”となるため、 $TICK_adj[1:0]$ のビット1の値に応じて、 $TICK$ の設定値が決められる。

【0046】

このような回路構成によれば、レジスタ42の $TICK_sync$ に論理“1”が書き込まれると、インクリメンタ54がキャリー出力をアサートしたときに、 $TICK_sync_out$ のパルスがアサートされる。キャリー出力のビット位置は、例えば、時刻調整プログラムが $TICK$ 同期処理を実施するために十分な時間が与えられるように、選択される。 $TICK_sync_out$ をアサートする必要がないときは、 $TICK_sync$ に論理“0”が設定される。

【0047】

また、レジスタ42のTICK. adj [1:0]は、2ビットのTICK調整用フィールドを持ち、*TS1-wrと*TICK-wrが論理“1”のとき、設定値に応じて、タイマ回路の状態とTICK. sync-in受信時の動作を制御する。

【0048】

TICK. adj [1:0] = 00の状態は定常状態に対応する。このとき、stopは論理“0”となり、countは論理“1”となるため、タイマ回路はカウント動作を行って、TICK値を更新する。この状態で、TICK. sync-inが論理“1”となっても、TICK. adj [1:0]の値は変化しないので、動作に影響はない。

【0049】

また、TICK. adj [1:0] = 01の状態は、システム時刻同期信号の伝播遅延時間を測定する測定状態に対応し、タイマ回路はカウント動作を行う。この状態で、TICK. sync-inが論理“1”となると、インバータ43の出力がTICK. adj [1:0]に設定されるので、TICK. adj [1:0] = 10となる。このとき、stopは論理“1”となり、countは論理“0”となるため、タイマ回路はカウント動作を停止する。

【0050】

また、TICK. adj [1:0] = 10の状態は停止状態に対応する。このとき、上述したように、タイマ回路はカウント動作を停止し、現在のTICK値を保持する。この状態で、TICK. sync-inが論理“1”となっても、TICK. adj [1:0]の値は変化しないので、動作に影響はない。

【0051】

また、TICK. adj [1:0] = 11の状態は、TICK同期処理を実施する同期状態に対応し、タイマ回路はカウント動作を停止する。この状態で、TICK. sync-inが論理“1”となると、インバータ43の出力がTICK. adj [1:0]に設定されるので、TICK. adj [1:0] = 00となる。これにより、定常状態に移行し、タイマ回路はカウント動作を再開する。

【 0 0 5 2 】

T I C K . a d j [1 : 0] の値とタイマ回路の状態の対応関係をまとめると、以下のようになる。

“ 0 0 ” : 定常状態 (カウント) 。

【 0 0 5 3 】

T I C K . s y n c - i n の受信時も変化なし。

“ 0 1 ” : 測定状態 (カウント) 。

T I C K . s y n c - i n の受信時に、カウント動作を停止。

【 0 0 5 4 】

“ 0 1 ” から “ 1 0 ” に変更される。

“ 1 0 ” : 停止状態 (停止) 。

T I C K . s y n c - i n の受信時も変化なし。

【 0 0 5 5 】

“ 1 1 ” : 同期状態 (停止) 。

T I C K . s y n c - i n の受信時に、カウント動作を再開。

“ 1 1 ” から “ 0 0 ” に変更される。

次に、図 4 を参照しながら、図 2 のパーティション制御回路 2 5 と T I C K 調整 T r e e 2 6 、 2 7 の具体例について説明する。図 4 においては、簡単のため、8 個の C P U (C P U # 0 ~ C P U # 7) からなるシステムが示されており、システムバス、スイッチ回路、および主記憶は省略されている。

【 0 0 5 6 】

また、パーティション制御回路 2 5 と T I C K 調整 T r e e 2 6 、 2 7 に対応する回路は、7 個の O R 回路 6 1 、 1 2 個の A N D 回路 6 2 、 4 個の O R 回路 6 3 、 2 個のインバータ回路 6 4 、およびパーティション制御レジスタ 6 5 を含む。

【 0 0 5 7 】

O R 回路 6 1 は、2 つの信号線 3 1 からの T I C K . s y n c - o u t の論理

和、または前段の2つのOR回路61の出力の論理和を求め、次段のOR回路61またはAND回路62に出力する。AND回路62は、OR回路61の出力とパーティション制御レジスタ65またはインバータ回路64の出力の論理和を求め、OR回路63に出力する。OR回路63は、3つのAND回路62の出力の論理積を求め、TICK. sync-inとして信号線32に出力する。

【0058】

また、インバータ回路64は、パーティション制御レジスタ65のビット0またはビット1の値を反転して、AND回路62に出力する。パーティション制御レジスタ65は、3つの制御ビットを有し、8個のCPUの論理的な分割方法を定義する。これらの制御ビットの値とパーティション構成の対応関係は、以下の通りである。ただし、()内のCPUが1つのパーティションを構成し、“x”は“0”と“1”のどちらでもよいことを意味する。

“000” : (CPU#0, CPU#1), (CPU#2, CPU#3),
 (CPU#4, CPU#5), (CPU#6, CPU#7)
“001” : (CPU#0, CPU#1), (CPU#2, CPU#3),
 (CPU#4, CPU#5, CPU#6, CPU#7)
“010” : (CPU#0, CPU#1, CPU#2, CPU#3),
 (CPU#4, CPU#5), (CPU#6, CPU#7)
“011” : (CPU#0, CPU#1, CPU#2, CPU#3),
 (CPU#4, CPU#5, CPU#6, CPU#7)
“1xx” : (CPU#0, CPU#1, CPU#2, CPU#3,
 CPU#4, CPU#5, CPU#6, CPU#7)

図4の回路構成によれば、パーティション制御レジスタ65にどのような値が設定されても、OR回路63は、同一パーティション内のすべてのCPUから出力されたTICK. sync-outの論理和を出力する。そして、OR回路63の出力は、TICK. sync-inとして、そのパーティション内のすべてのCPUに分配される。したがって、同一パーティション内のいずれかのCPU

がTICK. sync-outをアサートすれば、そのパーティション内のすべてのCPUのTICK. sync-inがアサートされる。

【0059】

時刻調整プログラムは、CPUがTICK. sync-outをアサートした時刻と、その信号がTICK調整Treeを経由してTICK. sync-inとして戻ってきた時刻の差に相当する時間を測定する。そして、測定された時間をシステム時刻同期信号の伝播遅延時間として、同一パーティション内のすべてのCPUの時刻調整プログラムに通知する。伝播遅延時間を通知された時刻調整プログラムは、それを用いてタイマ回路のTICK値を補正する。これにより、そのパーティション内のすべてのCPUのタイマ回路が同期する。

【0060】

図3のTICK調整回路では、タイマ回路を用いて伝播遅延時間を測定する。また、タイマ回路を既に調整済みの他のタイマ回路と同期させるために、タイマ回路のカウント動作を停止させることができる。時刻調整プログラムは、一旦、タイマ回路を停止状態にしてから、補正されたTICK値をTICK-wr[n:0]としてレジスタ53に書き込む。

【0061】

また、タイマ回路のカウント動作を再開させるために、調整済みのタイマ回路が更新開始信号の発生回路として用いられ、それを含むCPUのTICK. sync-outが更新開始信号出力として用いられる。また、被調整タイマ回路を含むCPUのTICK. sync-inが更新開始信号入力として用いられ、被調整タイマ回路は、更新開始信号入力に基づいてカウント動作を再開する。

【0062】

ところで、システム動作中は、TICK値を任意に変更することができないので、システム時刻同期信号の伝播遅延時間の測定は、システムの初期化時またはシステムの動的再構成時に行われる。

【0063】

システムの初期化時には、同一パーティション内の1つのCPU上の時刻調整プログラムが、そのパーティションのシステム時刻同期信号の伝播遅延時間を測

定する。そして、各CPUは、測定された時間を用いて時刻調整を行う。

【0064】

例えば、主記憶内にシステム共通パラメータを格納する領域を設けておき、初期化時に測定された伝播遅延時間を、共通パラメータの1つとして格納しておく。これにより、CPUは、タイマ回路の調整が必要になったときに、主記憶から伝播遅延時間を取得することができる。

【0065】

CPUの増設または交換を伴うシステムの動的再構成時には、追加されるCPUの時刻調整プログラムは、同一パーティション内のCPUから、伝播遅延時間を用いて補正されたTICK値を受け取る。次に、タイマ回路を停止させて、受け取ったTICK値をタイマ回路に設定し、TICK. sync-inの受信を待つ。そして、TICK. sync-inがアサートされるとともにカウント動作を再開することで、時刻調整を行う。

【0066】

また、追加されるCPUの時刻調整プログラムが単独で伝播遅延時間を測定することもできる。この場合、追加されるCPUは、基準となるTICK値を他のCPUから受け取り、そのTICK値を伝播遅延時間により補正して、タイマ回路に設定する。そして、TICK. sync-inがアサートされるとともにカウント動作を再開することで、時刻調整を行う。

【0067】

次に、図5および図6を参照しながら、TICK同期処理の具体例について説明する。

図5は、システムの初期化時におけるTICK同期処理の例を示している。この場合、同一パーティション内のCPUの1つがマスタCPUとなり、他のCPUがスレーブCPUとなる。まず、マスタCPU上でシステム初期化プログラムが実行され、初期化処理が終了すると、システム初期化プログラムは、マスタCPU上で動作する時刻調整プログラムに制御を移す。マスタCPUおよびスレーブCPUの処理手順は、以下の通りである。

【0068】

P1: マスタCPUの時刻調整プログラムは、すべてのスレーブCPUに対してTICK値(時刻)の同期処理開始を通知し、スレーブCPUからの応答を待つ。

【0069】

P2: 各スレーブCPUの時刻調整プログラムは、同期処理開始指示を受信し、マスタCPUに応答を通知する。

P3: マスタCPUの時刻調整プログラムは、すべてのスレーブCPUから応答を受信すると、タイマ回路からTICK値を読み出し、伝播遅延時間の測定処理を開始する。

【0070】

P4: マスタCPUの時刻調整プログラムは、読み出したTICK値から、TICK. sync-outがアサートされる(キャリー出力が出る)ときのTICK値を算定し、それを伝播遅延時間測定用のTICK値 T_0 とする。そして、*TS0-wr=0、TS0-wr[0]=1として、TICK. syncに論理“1”を設定し、*TS1-wr=0、TS1-wr[1:0]=01として、TICK. adjに“01”を設定する。これにより、マスタCPUのタイマ回路は、測定状態に設定される。

【0071】

P5: TICK値が T_0 に到達すると、マスタCPUは、TICK. sync-outをアサートする。TICK. sync-outは、TICK調整Treeを伝播し、時間tの後にTICK. sync-inとしてマスタCPUに戻る。

【0072】

P6: このとき、TICK. adj=01であるから、TICK. sync-inを受信すると、マスタCPUのタイマ回路は、TICK値の更新を停止するとともに、TICK. adj=10の状態に移る。マスタCPUの時刻調整プログラムは、TICK. adjの値の変化からタイマ回路の停止を検出し、測定処理を終了する。

【0073】

時刻調整プログラムは、測定終了直後に、マスタCPUのTICK値を読み出し、その値から T_0 を差し引くことで、伝播遅延時間 t を求める。次に、システム運用中のCPU増設に備えて、伝播遅延時間 t をシステム共通の記憶領域に一時的に格納し、TICK. adjに“00”を設定して、TICK値の更新を再開する。次に、読み出したTICK値から同期の基準となるTICK値 T_1 を算出し、 T_1 に t を加算してその値を補正する。そして、補正されたTICK値 $T_1 + t$ を、同期用TICK値としてすべてのスレーブCPUに通知する。

【0074】

P7：同期用TICK値を通知された各スレーブCPUの時刻調整プログラムは、 $*TICK - wr = 0$ 、 $TICK - wr[n:0] = (T_1 + t \text{ のバイナリコード})$ として、 $T_1 + t$ をTICK値に設定する。また、TICK. adjに“11”を設定して、タイマ回路を同期状態に設定し、同期準備完了の応答をマスタCPUに通知する。

【0075】

P8：マスタCPUの時刻調整プログラムは、すべてのスレーブCPUから応答を受信すると、処理を終了する。

その後、マスタCPUは、TICK値の更新を継続し、TICK値 T_1 に到達すると、TICK. sync-outをアサートする。TICK. sync-outは、TICK調整Treeを伝播し、時間 t 後の時刻 $T_1 + t$ に、TICK. sync-inとしてすべてのスレーブCPUに到達する。

【0076】

TICK. sync-inを受信した各スレーブCPUのタイマ回路は、あらかじめ設定されているTICK値 $T_1 + t$ から更新を再開し、TICK. adj = 00の状態（定常状態）に遷移することで、TICK同期処理を終了する。こうして、マスタCPUのタイマ回路が $T_1 + t$ をカウントしたとき、すべてのスレーブCPUのタイマ回路も、同じTICK値からカウントを開始する。

【0077】

また、図6は、プロセッサ増設時におけるTICK同期処理の例を示している。この場合、既存のプロセッサの1つがマスタCPUとなり、追加されるCPU

がスレーブCPUとなる。OSは、システムの全処理を終了させた後、マスタCPU上で動作する時刻調整プログラムに制御を移し、スレーブCPUの情報（CPU番号等）を時刻調整プログラムに通知する。

【0078】

図6のP11～P13の処理手順については、図5のP1～P3と同様である。その後のマスタプロセッサおよびスレーブプロセッサの処理手順は、以下の通りである。

【0079】

P14：マスタCPUの時刻調整プログラムは、P13で読み出したTICK値から、TICK. sync-outがアサートされときのTICK値を算定し、それを同期の基準となるTICK値 T_1 とする。次に、システム共通の記憶領域から、システム初期化時に格納された伝播遅延時間 t を取得し、 T_1 に t を加算して T_1 を補正する。そして、補正されたTICK値 $T_1 + t$ を、同期用TICK値としてスレーブCPUに通知する。

【0080】

そして、TICK. syncに論理“1”を設定して、TICK. sync-outをアサートする準備を行う。このとき、TICK. adjは“00”のままで変更されないので、マスタCPUのタイマ回路は、停止せずにTICK値の更新を継続する。

【0081】

P15：同期用TICK値を通知されたスレーブCPUの時刻調整プログラムは、 $T_1 + t$ を更新再開時刻としてTICK値に設定する。また、TICK. adjに“11”を設定して、タイマ回路を同期状態に設定し、同期準備完了の応答をマスタCPUに通知する。

【0082】

P16：マスタCPUの時刻調整プログラムは、スレーブCPUから応答を受信すると、処理を終了して、制御をOSに戻す。

その後のマスタCPUとスレーブCPUの動作は、図5の場合と同様である。こうして、マスタCPUのタイマ回路が $T_1 + t$ をカウントしたとき、スレーブ

CPUのタイマ回路も、同じTICK値からカウントを開始する。

【0083】

以上説明した実施形態においては、主として、対称型マルチプロセッサシステムにおける時刻調整について述べたが、本発明は任意のマルチプロセッサシステムに適用することができる。

【0084】

また、上述の実施形態においては、TICK、sync-outを生成するためのタイミング信号として、タイマ回路のキャリー出力を用いているが、これを他のタイミング信号に置き換えてもよい。また、タイマ回路がカウント動作を再開するための信号としてTICK、sync-inを用いているが、これを他のタイミング信号に置き換えてもよい。

【0085】

さらに、上述の実施形態においては、タイマ回路を用いてシステム時刻同期信号の伝播遅延時間を測定しているが、ソフトウェア等の他の方法で時間を測定してもよい。

(付記1) マルチプロセッサシステムにおける複数のプロセッサのタイマを調整するタイマ調整システムであって、

前記マルチプロセッサシステムの時刻同期信号を発生する発生手段と、

前記時刻同期信号を出力する出力手段と、

前記出力手段から出力され、前記マルチプロセッサシステム内を伝播して戻ってきた前記時刻同期信号を、入力する入力手段と、

前記出力手段が前記時刻同期信号を出力してから前記入力手段が該時刻同期信号を入力するまでの時間を測定する測定手段と、

測定された時間を、前記時刻同期信号が前記マルチプロセッサシステム内の2つのプロセッサ間を伝播する時間として用いて、前記複数のプロセッサのタイマのうち少なくとも1つ以上のタイマの時刻情報を補正し、該複数のプロセッサのタイマを同期させる同期手段と

を備えることを特徴とするタイマ調整システム。

(付記2) 前記発生手段は、前記複数のプロセッサのタイマのうちの1つが生

成する信号を用いて、前記時刻同期信号を発生することを特徴とする付記 1 記載のタイマ調整システム。

(付記 3) 前記測定手段は、前記複数のプロセッサのタイマのうちの 1 つを用いて前記時刻同期信号が伝播する時間を測定することを特徴とする付記 1 記載のタイマ調整システム。

(付記 4) 前記同期手段は、前記複数のプロセッサのタイマのうちの基準タイマを制御するソフトウェア命令を実行する第 1 のソフトウェア手段と、補正対象のタイマを制御するソフトウェア命令を実行する第 2 のソフトウェア手段とを含み、該基準タイマの時刻情報と前記測定された時間を用いて、該補正対象のタイマの時刻情報を補正することを特徴とする付記 1 記載のタイマ調整システム。

(付記 5) 前記入力手段が前記時刻同期信号を入力したとき、前記補正対象のタイマのカウント動作を開始させる開始信号を生成する生成手段をさらに備え、前記第 2 のソフトウェア手段は、該補正対象のタイマを停止させて、補正された時刻情報を該補正対象のタイマに格納し、該補正対象のタイマを、該開始信号に基づいてカウント動作を開始する状態に設定することを特徴とする付記 4 記載のタイマ調整システム。

(付記 6) 前記測定手段は、プロセッサの追加を伴う前記マルチプロセッサシステムの再構成時に、前記時刻同期信号が伝播する時間を測定し、前記第 2 のソフトウェア手段は、追加されるプロセッサのタイマを前記補正対象のタイマとして制御することを特徴とする付記 5 記載のタイマ調整システム。

(付記 7) 前記測定された時間を格納する格納手段をさらに備え、前記測定手段は、前記マルチプロセッサシステムの初期化時に、前記時刻同期信号が伝播する時間を測定して、該格納手段に格納し、前記同期手段は、前記補正対象のタイマの時刻情報を補正するときに、格納された時間を取得することを特徴とする付記 1 記載のタイマ調整システム。

(付記 8) マルチプロセッサシステムにおける複数のプロセッサのタイマを調整するタイマ調整システムであって、

前記マルチプロセッサシステムの時刻同期信号を発生する発生手段と、

前記時刻同期信号を同期出力として出力する出力手段と、

同期入力を入力する入力手段と
を前記複数のプロセッサの各々に設け、

前記複数のプロセッサから出力される複数の同期出力の論理和信号を生成し、
該論理和信号を前記同期入力として前記複数のプロセッサに分配する分配手段と

、
前記複数のプロセッサのうちの 1 つが前記同期出力を出力してから前記同期入
力を受け取るまでの時間を測定する測定手段と、

測定された時間を、前記論理和信号が伝播する時間として用いて、前記複数の
プロセッサのタイマのうち少なくとも 1 つ以上のタイマの時刻情報を補正し、該
複数のプロセッサのタイマを同期させる同期手段と

を備えることを特徴とするタイマ調整システム。

(付記 9) 前記マルチプロセッサシステムを複数のパーティションに分割して
、各パーティションを対称型マルチプロセッサシステムとして動作させる制御手
段をさらに備え、前記分配手段は、各パーティションに属するプロセッサから出
力される同期出力の論理和信号を生成し、該論理和信号を同期入力として該パー
ティションに属するプロセッサに分配し、前記同期手段は、各パーティションに
属するプロセッサのタイマを同期させることを特徴とする付記 8 記載のタイマ調
整システム。

【 0 0 8 6 】

【発明の効果】

本発明によれば、マルチプロセッサシステムにおいて、単純で安価な少量のハ
ードウェアを追加するだけで、ソフトウェアによる時刻調整を支援し、極めて誤
差の小さい時刻調整を行うことができる。また、システムのプロセッサ数に依存
しない時刻調整が実現される。

【図面の簡単な説明】

【図 1】

本発明のタイマ調整システムの原理図である。

【図 2】

マルチプロセッサシステムの構成図である。

【図 3】

T I C K 調整回路の構成図である。

【図 4】

パーティション制御回路と T I C K 調整 T r e e の構成図である。

【図 5】

第 1 の T I C K 調整を示す図である。

【図 6】

第 2 の T I C K 調整を示す図である。

【符号の説明】

- 1 1 発生手段
- 1 2 出力手段
- 1 3 入力手段
- 1 4 測定手段
- 1 5 同期手段
- 1 6 分配手段
- 2 1 C P U
- 2 2 システムバス
- 2 3 スイッチ回路
- 2 4 主記憶
- 2 5 パーティション制御回路
- 2 6、2 7 T I C K 調整 T r e e
- 3 1、3 2、3 3、3 4 信号線
- 4 0、4 1、5 5 マルチプレクサ
- 4 2 T I C K 同期レジスタ
- 4 3、4 4、4 7、4 9、5 2、6 4 インバータ回路
- 4 5、4 6、5 0、5 1、5 6、6 2 A N D 回路
- 4 8 N A N D 回路
- 5 3 T I C K レジスタ
- 5 4 インクリメンタ

6 1、6 3 O R 回路

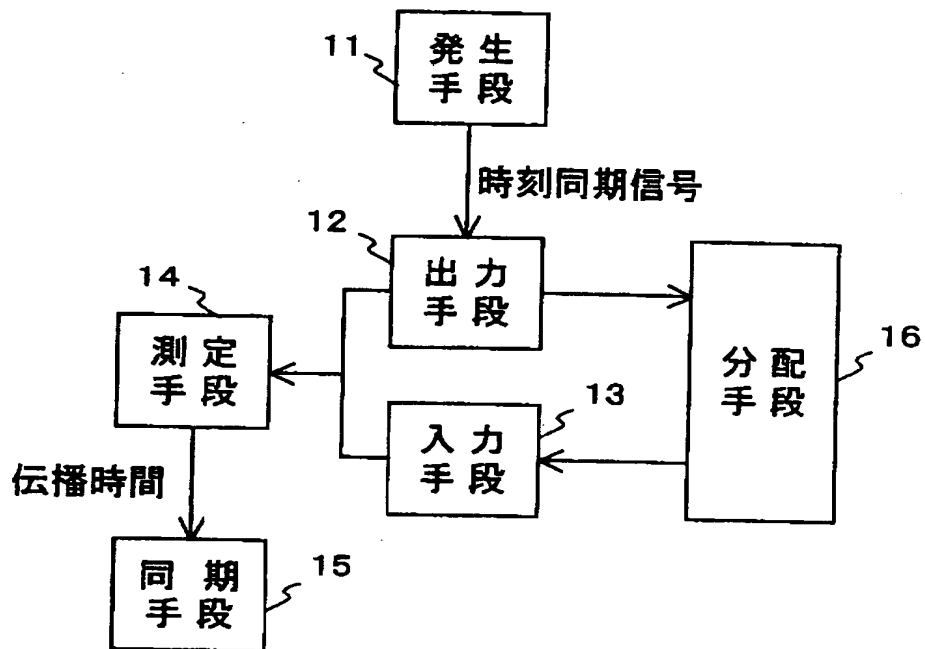
6 5 パーティション制御レジスタ

【書類名】

図面

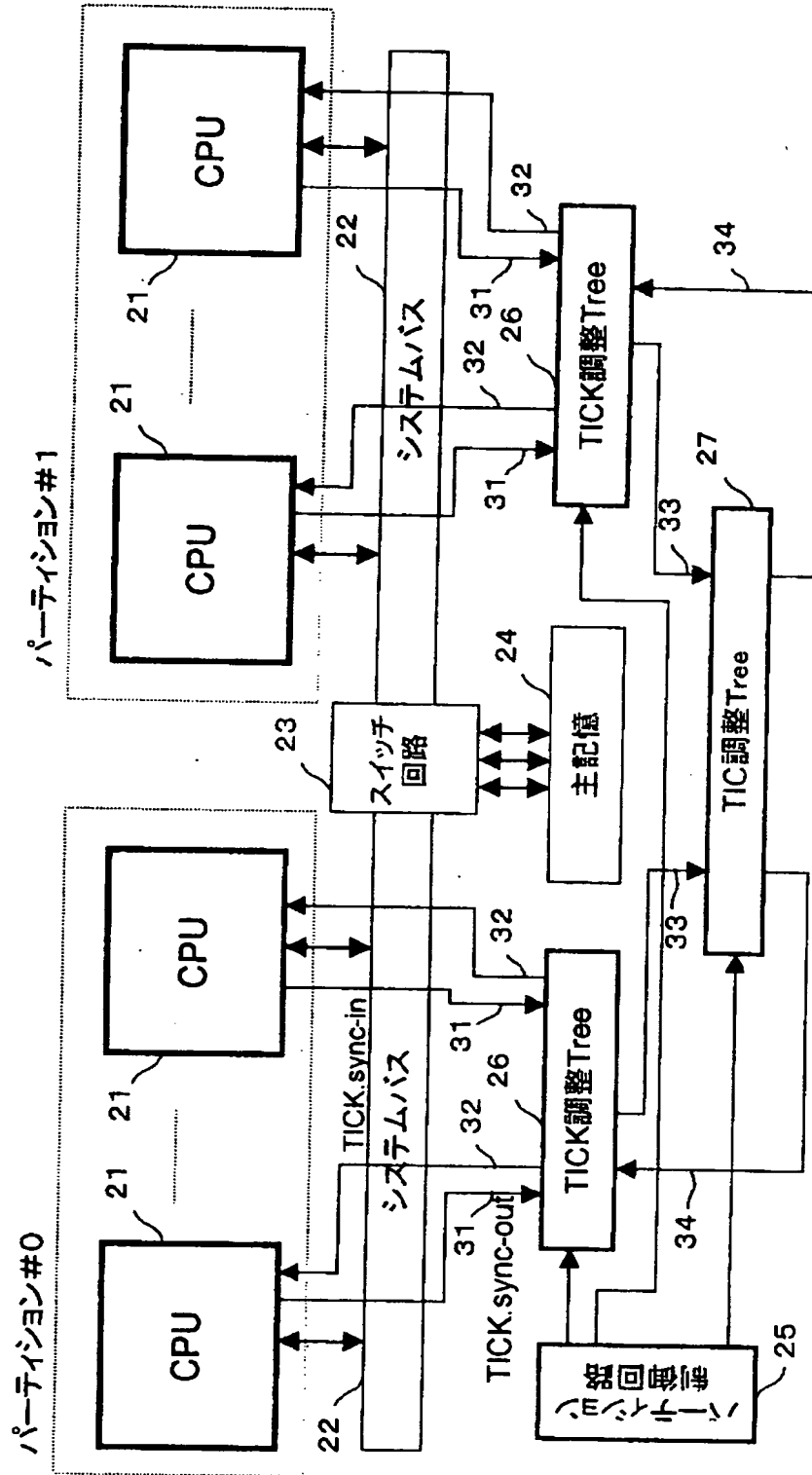
【図 1】

本 発 明 の 原 理 図



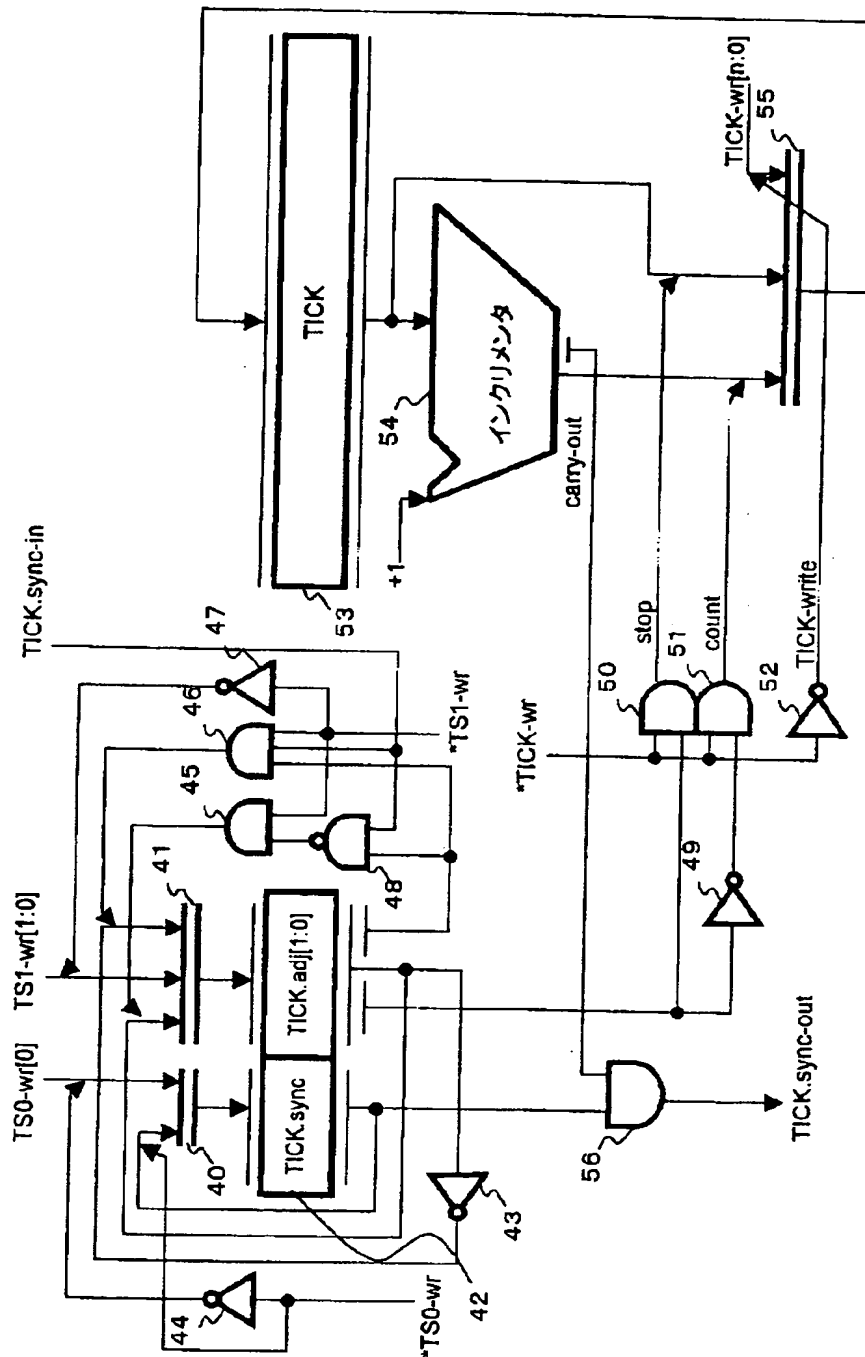
【図 2】

マルチプロセッサシステムの構成図



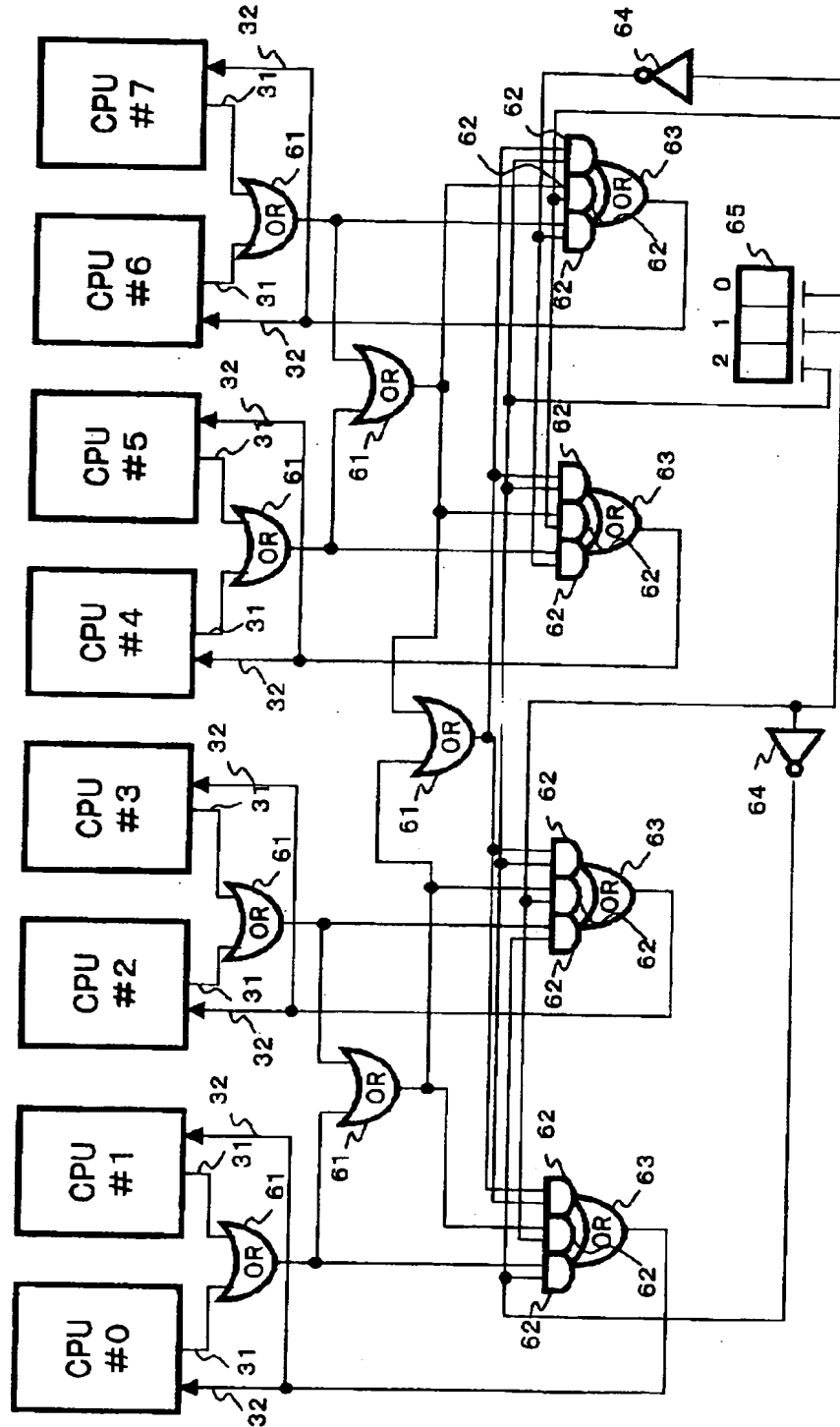
【図 3】

TICK 調整回路の構成図



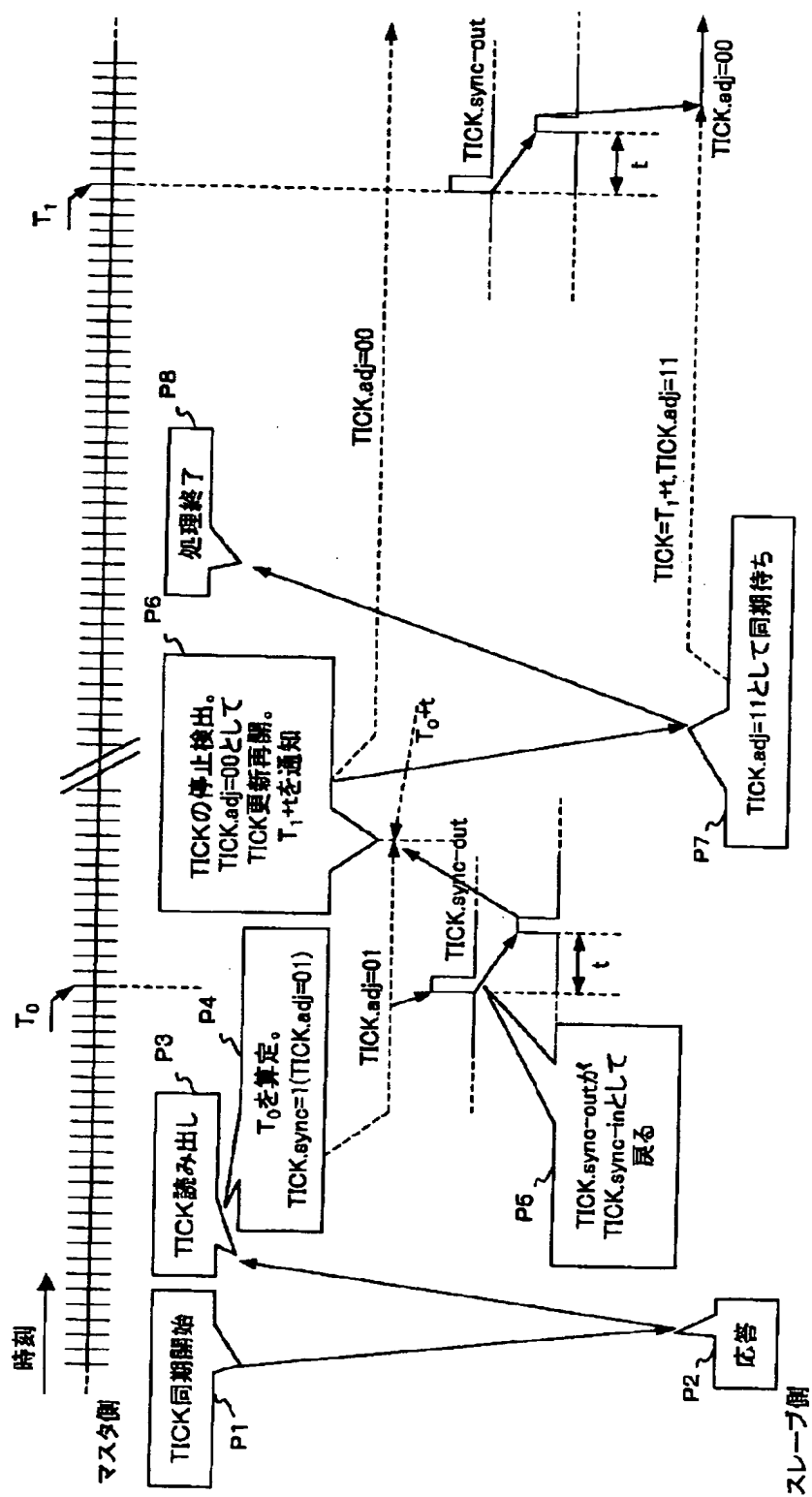
【図 4】

パーティション制御回路とTCK調整Treeの構成図



【図 5】

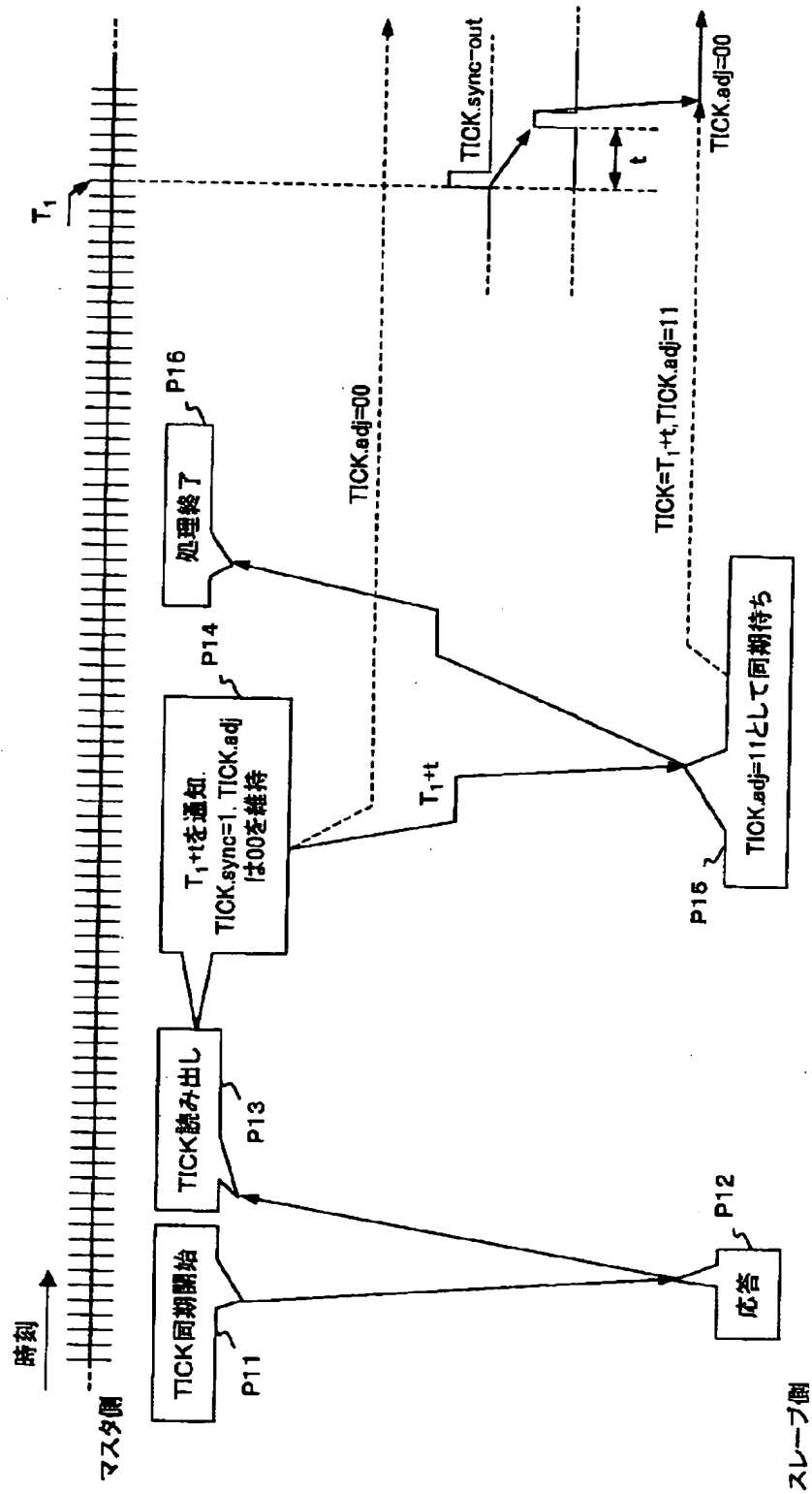
第 1 の T I C K 調 整 を 示 す 図



特 2 0 0 0 - 2 3 3 8 1 1

【図 6】

第 2 の T I C K 調 整 を 示 す 図



【書類名】 要約書

【要約】

【課題】 マルチプロセッサシステムにおいて、プロセッサ間の時刻調整をより正確に行うことが課題である。

【解決手段】 レジスタ42のTICK. syncに論理“1”が書き込まれると、インクリメンタ54がキャリー出力をアサートしたときに、TICK. sync-outがアサートされ、時刻同期信号として他のプロセッサに伝達されるとともに、TICK. sync-inとして戻ってくる。この信号の伝播遅延時間を測定し、測定された時間を用いて、各プロセッサのレジスタ53に保持されたTICK値を補正する。

【選択図】 図3

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社